UTILITY PATENT APPLICATION TRANSMITTAL

(Only for new nonprovisional applications under 37 CFR 1.53(b))

Attorney Docket No.	P98,3[redacted]
First Named Inventor or Application Identifier	
Karlheinz Dorn et al,	
Express Mail Label No:	# EL121676562US

ADDRESS TO: **Assistant Commissioner for Patents**
Box Patent Application
Washington, DC 20231

jc585 U.S. PTO
09/215732
12/18/98

APPLICATION ELEMENTS

See MPEP chapter 600 concerning utility patent application contents.

1. X Specification [Total Pages 27]
2. X Drawing(s) (35USC 113) [Total Pages 10]
3. X Declaration and Power of Attorney [Total Pages 3]
   a. __ Newly executed(original copy)
   b. __ Copy from prior application (37CFR 1.63(d)) *(for continuation/divisional with Box 14 completed)*
      [Note Box 4 Below]
      i. __ DELETION OF INVENTOR(S) Signed statement attached deleting Inventor(s) named in the prior application, see 37 CFR 1.63(d)(2) and 1.33(b).
4. __ Incorporation By Reference (usable if Box 3b is checked) The entire disclosure of the prior application, from which a copy of the oath or declaration is supplied under Box 3b, is considered as being part of the disclosure of the accompanying application and is hereby incorporated by reference therein.

ACCOMPANYING APPLICATION PARTS

5. __ Assignment Papers (cover sheet & documentation)
6. X Letter under 37 CFR 1.41(c).
7. __ English Translation Document *(if applicable)*
8. __ Information Disclosure Statement (IDS)/PTO-1449 __ Copies of IDS Citations
9. __ Preliminary Amendment
10. X Return Receipt Postcard (MPEP 503) *(Should be specifically itemized)*
11. __ Small Entity Statement(s) __ Statement filed in prior application, Status still proper and desired
12. __ Certified Copy of Priority Document(s).
13. __ Other:

14. If a CONTINUING APPLICATION, *check appropriate box and supply the requisite information:*

__ Continuation __ Divisional __ Continuation-in-part (CIP) __ of prior application No:____/____

CLAIMS AS FILED

(1) FOR	(2) NUMBER FILED	(3) NUMBER EXTRA	(4) RATE	(5) BASIC FEE $760.00
TOTAL CLAIMS 20	5			
INDEPENDENT CLAIMS 3	1			
	ANY MULTIPLE DEPENDENT CLAIMS? ()YES (X) NO			
			TOTAL FILING FEE ->	$760.00

X The Commissioner is hereby authorized to charge any additional fees which may be required in connection with this application, or credit any overpayment to ACCOUNT NO. 08-2290. A duplicate copy of this sheet is enclosed.

X A check in the amount of $760.00 to cover the filing fee is enclosed.

15. CORRESPONDENCE ADDRESS

HILL & SIMPSON
A Professional Corporation
233 South Wacker Drive - 85th Floor Sears Tower
Chicago, Illinois 60606
Telephone (312) 876-0200 - Fax (312) 876-0898

SIGNATURE: Daniel L Mehler **DATE:** December 18, 1998

491/1044:1190

CERTIFICATE OF MAILING

"Express Mail" Mailing Label Number **EL 121676562 US**

Date of Deposit: December 18, 1998

I hereby certify that this correspondence is being deposited with the United States Postal "Express Mail Post Office to Addressee" service under 37 CFR 1.10 on the date indicated above and is addressed to:

BOX PATENT APPLICATION
Assistant Commissioner for Patents
Washington, D.C. 20231

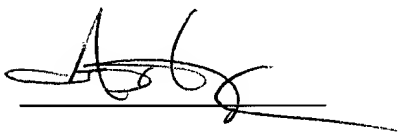
Case Number: P98,3059

Inventors: Karlheinz Dorn et al.

Patent Application entitled:

"INTERPRETIVE NETWORK DAEMON IMPLEMENTED BY GENERIC MAIN OBJECT"

Signature of person mailing documents and fee



HILL & SIMPSON

A PROFESSIONAL CORPORATION
ATTORNEYS AND COUNSELORS AT LAW
CHICAGO, ILLINOIS 60606

JOHN D. SIMPSON *
LEWIS T. STEADMAN
JAMES A. MOEHLING
DENNIS A. GROSS
ROBERT M. BARRETT
STEVEN H. NOLL
KEVIN W. GUYNN
SCOTT W. PETERSEN
ROBERT M. WARD
BRETT A. VALIQUET
GEORGE C. SUMMERFIELD
LEWIS T. STEADMAN, JR.
EDWARD A. LEHMAN
DAVID R. METZGER
TODD S. PARKHURST
JOHN R. NYWEIDE
JAMES D. HOBART
MELVIN A. ROBINSON
JOHN R. GARRETT
C. GRANT McCORKHILL

PAULA J. KELLY
JOHN W. CORNELL

ROBERT J. DEPKE
PATRICIA A. KANE
JOSEPH P. REAGEN
MICHAEL R. HULL
MICHAEL S. LEONARD
WILLIAM E. VAUGHAN

JAMES VAN SANTEN
J. ARTHUR GROSS
MARVIN MOODY

DOLORES K. HANNA
SPECIAL TRADEMARK COUNSEL

CHICAGO OFFICE
85TH FLOOR SEARS TOWER
CHICAGO, ILLINOIS 60606
TELEPHONE (312) 876-0200
FACSIMILE (312) 876-0898
INTERNET: counsel@hillfirm.com

WASHINGTON OFFICE
SUITE 1004-BLDG. 1
2001 JEFFERSON DAVIS HIGHWAY
CRYSTAL CITY
ARLINGTON, VIRGINIA 22202
TELEPHONE (703) 415-1515

* MUNICH OFFICE
FRANZ-JOSEPH STRASSE 38
D-80801 MUNICH, GERMANY
49-89-3840720

December 18, 1998

Assistant Commissioner for Patents
Washington, D.C. 20231

Re: U.S. patent application for DORN et al., "INTERPRETIVE NETWORK DAEMON IMPLEMENTED BY GENERIC MAIN OBJECT", Our File No. P98,3059

S I R:

Under the provisions of 37 CFR 1.41 (c), I am filing the attached patent application on behalf of Karlheinz Dorn et al. an unexecuted Declaration being submitted herewith, and request that the application papers be assigned a serial number and filing date pursuant to the provisions of 1.53(b) and 1.53(d).

Respectfully submitted,

Melvin A. Robinson (Reg. No. 31,870)

1044/1072
Enclosures

```
/*| Compilation unit ----------------------------------------------------------

   Component       : CSA -

   Name            : CsaGenericMainBas.h

   Author          : (D. Quehl, BNEP11, +49 9131 84 2430); Siemens AG B Med CSA

   Language        : C/C++

   Creation Date   : 19-JUN-1996

   Test State      :

   Description     : Definition of Generic Main functionality.

   Requirement Key : os_os_process_mgr

                Copyright (C) Siemens AG 1995 All Rights Reserved

-----------------------------------------------------------------------------*/
/*| END */
#ifndef CSAGENERICMAINBAS_H
#define CSAGENERICMAINBAS_H

#pragma comment ( exestr,"@(#) CsaGenericMainBas.h 1.2 (10 Jul 1996)  : Definition Generic Main Base class" )

#include <ace/OS.h>
#include <ace/Task.h>
#include <ace/Synch.h>
#include <ace/Synch_T.h>

#include <CsaCommon/CsaDefs.h>
#include  <osc/CsaOscNlsMessageId.h>



/*| Data type -----------------------------------------------------------------

   Name        : CSA_STATUS - macro for Msc component

   Description : Macro for error Handling
                 -- uses status as Input
                 -- hides setting of Line and File

-----------------------------------------------------------------------------*/
#ifndef CSA_ERH_DISABLE
#include  <erh/CsaStatus.h>
#define   CSA_STATUS_gmain(status) ! csaStatus->set(status,__FILE__,__LINE__);|
#else
#define   CSA_STATUS_gmain(status) ;
#endif    // #ifdef CSA_ERH_DISABLE
#define CSA_STATUS_GMAIN(status) CSA_STATUS_gmain(CSA_OSC_MSC_##status)
```



```
/*| END Data type */


/*| Data type -----------------------------------------------------------------

  Name        : CSA_TRACE_IN - Macro redefinition.

  Description : Macro redefinition that disables method trace.

-----------------------------------------------------------------------------*/
#ifndef CSA_TRACE_DISABLE
#include        <erh/CsaTrace.h>
#else
#define   CSA_TRACE_IN(a)
#endif  // #ifndef CSA_TRACE_DISABLE

/*| END Data type */




/*| Class ---------------------------------------------------------------------

   Name           : CsaGenericMainBas

   Description    : CsaGenericMainBas provides the gereic main program
                    functionality. CsaGenericMainBas is derived from
                    ACE_Task<ACE_MT_SYNCH> to let the dispatch loop
                    run in either, the main (caller) thread or a
                    separate thread (task's svc method).

-----------------------------------------------------------------------------*/
class CsaGenericMainBas : public ACE_Task<ACE_MT_SYNCH> {

public:

  // Constructor/destructor
  CsaGenericMainBas(void);
  CsaGenericMainBas(int argc, char *argv[]);
  ~CsaGenericMainBas(void) {
    delete thrMgr_;
  };

  // Store commandline  arguments if instantiated using the default
  // constructor.
  void setArgs (int argc, char *argv[]) {
    argc_ = argc;
    argv_ = argv;
  }

  // handle SIGINT
  virtual int handle_signal (int signum);

  // Perform initialization steps for CsaGenericMainBas
```

```
  virtual int open (void *p = 0);

  // Invoke dispatch loop [a]synchronously.
  virtual int dispatchLoop(bool async = false);

  // synchronize on dispatcher termination
  virtual void sync(void);

  // break the loop
  virtual int endLoop(void);

  // hook methods that can be overloaded to insert functionality before
  // the loop starts and immediatly after the loop finished.
  virtual int loopHook1(void) { return 0; }
  virtual int loopHook2(void) { return 0; }

  // hook methods that can be overloaded to insert functionality prior
  // to daemon open and after daemon close.
  virtual int openHook(void) { return 0; }
  virtual int closeHook(void) { return 0; }

private:

  // The implementation of the main dispatch loop.
  virtual int svc (void);

  // The following methods are inherited from ACE_Task which defines
  // them as pure virtual methods.
  // They are not needed here and implemented returning instant error.
  // Subclasses of CsaGenericMainBas may overload them.
  virtual int put(ACE_Message_Block *, ACE_Time_Value * =0) {
    return -1;
  }
  virtual int close (unsigned long flags = 0) {
    return 0;
  }
  virtual int handle_input (ACE_HANDLE fd = ACE_INVALID_HANDLE) {
    return -1;
  }

  // The service configuration daemon
  ACE_Service_Config  daemon_;

  // Copy of the command line arguments
  int argc_;
  char **argv_;

  // Indicator for initialization state
  bool initialized_;

  // API lock that prevents loop from beeing called multiply
  // and synchronizes the termination of the thread that runs
  // the loop asynchronously with the caller thread.
  ACE_Thread_Mutex lock_;

  // provide ACE_task a thread manager
```



```
  ACE_Thread_Manager  *thrMgr_;

};

/*] END Class */


#endif  //ifndef CSAGENERICMAINBAS_H


/*[ Change header ---------------------------------------------------------

   Date         : 26. Jun 1996
   Version      : -
   Charm        : -
   Author       : D. Quehl (Qu); Siemens AG B Med CSA

   Description  : Initial release

]-----------------------------------------------------------------------*/


/*[ Change header ---------------------------------------------------------

   Date         : 07. Jul 1996
   Version      : 1.1
   Charm        : 2041
   Author       : D. Quehl (Qu); Siemens AG B Med CSA

   Description  : Changes in respect to code review component
                  "osc - generic main - 03.jul.96 "

]-----------------------------------------------------------------------*/
```

```
/*| Compilation unit ----------------------------------------------------

   Component       : CSA -

   Name            : CsaGenericMainBas.cpp

   Author          : (D. Quehl, BNEP11, +49 9131 84 2430); Siemens AG B Med CSA

   Language        : C/C++

   Creation Date   : 19-JUN-1996

   Test State      :

   Description     : Implementation of the Generic Main functionality.

   Requirement Key : os_os_process_mgr, os_os_test_testability

             Copyright (C) Siemens AG 1995 All Rights Reserved

-----------------------------------------------------------------------------*/
/*| END */

#pragma comment ( exestr, "@(#) CsaGenericMainBas.cpp 1.3 (10 Jul 1996)  : Implementation Generic Main Base class" )

#include <osc/CsaGenericMainBas.h>

/*| Method --------------------------------------------------------------

   Name            : CsaGenericMainBas::CsaGenericMainBas

   Description     : Default constructor for CsaGenericMainBas.
                     Prior to starting the dispatch loop the
                     command line arguments must be stored using
                     the setArgs() method.
                     The constructor calls the open() method.

   Return          : none

-----------------------------------------------------------------------------*/
CsaGenericMainBas::CsaGenericMainBas(
                            void     // takes no arguments
                            )

/*| END Method */

  : argc_(0), argv_(0), initialized_(false)
{
  CSA_TRACE_IN ((CSA_OSC, "int CsaGenericMainBas::CsaGenericMainBas"));
  this->open();

/*| Method --------------------------------------------------------------
```



```
   Name            : CsaGenericMainBas::CsaGenericMainBas

   Description     : Constructor for CsaGenericMainBas.
                     Stores the command line arguments and calls the open()
                     method.

   Return          : none

-----------------------------------------------------------------------------*/
CsaGenericMainBas::CsaGenericMainBas(
                            int argc,       // argument count
                            char *argv[]    // argument vector
                            )

/*| END Method */

  : initialized_(false)
{
  CSA_TRACE_IN ((CSA_OSC, "int CsaGenericMainBas::CsaGenericMainBas"));

  this->setArgs(argc,argv);
  this->open();
}


/*| Method --------------------------------------------------------------

   Name            : CsaGenericMainBas::open

   Description     : Perform initialization steps for CsaGenericMainBas.
                     - Create our own thread manager

   Return          : 0                       success
                     -1                      error

-----------------------------------------------------------------------------*/
int CsaGenericMainBas::open (
                            void *p         // pointer for arbitrary use
                            )

/*| END Method */

{
  CSA_TRACE_IN ((CSA_OSC, "int CsaGenericMainBas::open"));

  if ((this->thrMgr_ = new ACE_Thread_Manager) == 0) {
    CSA_STATUS_GMAIN(NOMEM_NEW);
    return -1;
  }
  this->thr_mgr(thrMgr_);                   // pass thread manager to task

  return 0;
}

/*| Method --------------------------------------------------------------
```

```
*
   Name            : CsaGenericMainBas::handle_signal

   Description     : The handle_signal() method is called on an incoming
                     SIGINT.
                     handle_signal() calls endLoop() to terminate dispatching.

   Return          : 0                         success
                     -1                        error

------------------------------------------------------------------------*/
int CsaGenericMainBas::handle_signal (
                                int signum // signal number that caused invokation
                                )

/*] END Method */

{
  CSA_TRACE_IN ((CSA_OSC, "int CsaGenericMainBas::handle_signal"));

  if (this->thrMgr_ != 0)
    return this->endLoop();

  CSA_STATUS_GMAIN(NOMEM_NEW);
  return -1;
}

/*[ Method ---------------------------------------------------------------

   Name            : CsaGenericMainBas::svc

   Description     : The svc() method implementation of the main dispatch loop.
                     After starting the ACE_Service_Config daemon with the
                     stored command line arguments the first hook method is
                     called and - if this hook returns success - the dispatch
                     loop invoked.
                     Immediately after returning from the dispatch loop the
                     second hook method is called to perform subclass specific
                     cleanup actions while the ACE_Service_Config is still up.

   Return          : 0                         success
                     -1                        error

------------------------------------------------------------------------*/
int CsaGenericMainBas::svc (
                        void                // takes no arguments
                        )

*] END Method */

{
  CSA_TRACE_IN ((CSA_OSC, "int CsaGenericMainBas::svc"));

  // return error if thread manager creation failed in open()
  if (this->thrMgr_ == 0) {
    CSA_STATUS_GMAIN(NOMEM_NEW);
```



```
    return -1;
  }

  // Lock the access to the dispatch loop
  ACE_TSS_Guard<ACE_Thread_Mutex> mon (this->lock_);

   // make us owner of the reactor
  ACE_Service_Config::reactor()->owner(ACE_Thread::self());

 // invoke the startHook
  if (this->openHook() == -1)
    return -1;

  // Start the daemon
  ACE_SEH_TRY {
    if (this->daemon_.open (this->argc_, this->argv_) == -1) {
      CSA_STATUS_GMAIN(NO_REACTOR);
      return -1;
    }
  }
  ACE_SEH_EXCEPT(EXCEPTION_EXECUTE_HANDLER) {
    cout << "Caught structured exception while opening Service_Config daemon" << endl;
  }

  // Register this to the reactor to receive SIGINT
  if (ACE_Service_Config::reactor()->register_handler (SIGINT, this) == -1) {
    CSA_STATUS_GMAIN(REGISTER_HANDLER);
    (void)this->daemon_.close();
    return -1;
  }

  // Call the first hook; fall through if hook returns an error
  if (this->loopHook1() >= 0) {
    initialized_ = true;
    ACE_SEH_TRY {
      ACE_Service_Config::run_event_loop ();
    }
    ACE_SEH_EXCEPT(EXCEPTION_EXECUTE_HANDLER) {
      cout << "Caught structured exception while running Service_Config event loop" << endl;
    }
    (void)this->loopHook2();
  }

  int status;

  // Close down the services of daemon
  ACE_SEH_TRY {
    status = this->daemon_.close_svc();
  }
  ACE_SEH_EXCEPT(EXCEPTION_EXECUTE_HANDLER) {
    cout << "Caught structured exception while closing down Service_Config daemon" << endl;
  }

  // invoke the close hook
  (void)this->closeHook();
```

```
  // Close down the memory of daemon
  status = this->daemon_.close_mem();

  initialized_ = false;

  return status;
}

/*| Method ----------------------------------------------------------------

   Name           : CsaGenericMainBas::dispatchLoop

   Description    : Invoke dispatch loop either synchronously (in the context
                    of the caller thread) or asynchronously (using
                    ACE_TASK::activate() to create a separate thread).

   Return         : 0                        success
                    -1                       error

----------------------------------------------------------------------------*/
int CsaGenericMainBas::dispatchLoop(
                              bool async   // flag for asynchronous invokation
                              )

/*| END Method */

{

  // return error if thread manager creation failed in open()
  if (this->thrMgr_ == 0) {
    CSA_STATUS_QMAIN(NOMEM_NEW);
    return -1;
  }

  // check whether initialization was already set
  if (!initialized_) {

    // for synchronous operation just invoke svc method in the context
    // of the caller's thread.
    if (!async)
      return this->svc();

    // for asynchronous operation invoke svc method in the context
    // of a thread created by activate().
    else
      return this->activate();
  }

  // Return error if invoked twice.
  return -1;
}


/*| Method ----------------------------------------------------------------

   Name           : CsaGenericMainBas::endLoop
```



```
   Description    : Stop dispatching.

   Return         : 0                        success
                    -1                       error

----------------------------------------------------------------------------*/
int CsaGenericMainBas::endLoop(
                              void         // takes no arguments
                              )

/*| END Method */

{
  ACE_Service_Config::end_event_loop();
  return 0;
}


/*| Method ----------------------------------------------------------------

   Name           : CsaGenericMainBas::sync

   Description    : Synchronize on dispatcher thread finished

   Return         : none

----------------------------------------------------------------------------*/
void CsaGenericMainBas::sync(
                              void         // takes no arguments
                              )

/*| END Method */

{
  if (this->thrMgr_ != 0)
    this->thrMgr_->wait();
  ACE_Thread::exit(0);
}


/*| Change header ---------------------------------------------------------

   Date         : 26. Jun 1996
   Version      : -
   Charm        : -
   Author       : D. Quehl (Qu); Siemens AG B Med CSA

   Description  : Initial release

|---------------------------------------------------------------------------*/


/*| Change header ---------------------------------------------------------

   Date         : 07. Jul 1996
```

```
   Version     : 1.1
   Charm       : 2041
   Author      : D. Quehl (Qu); Siemens AG B Med CSA

   Description : Changes in respect to code review component
                 "osc - generic main - 03.jul.96 "

|-------------------------------------------------------------------------*/
```

```
/*| Compilation unit ----------------------------------------------------

   Component       : CSA -

   Name            : CsaGenericMain.h

   Author          : (D. Quehl, BNEP11, +49 9131 84 2430); Siemens AG B Med CSA

   Language        : C/C++

   Creation Date   : 19-JUN-1996

   Test State      :

   Description     : Add on to Generic Main functionality.
                     Definition of library startup/rundown functionality.

   Requirement Key : os_os_process_mgr

                Copyright (C) Siemens AG 1995 All Rights Reserved

-----------------------------------------------------------------------------*/
/*| END */
#ifndef CSAGENERICMAIN_H
#define CSAGENERICMAIN_H

#pragma comment ( exestr,"@(#) CsaGenericMain.h 1.4 (10 Jul 1996)  test: Definition Generic Main" )

#include <osc/CsaGenericMainBas.h>

// Header files of Singletons
#include <osc/CsaSesam.h>
#include <osc/CsaMscCommu.h>
#include <osc/CsaOsSystem.h>
#include <osc/CsaPmgtCmdIf.h>


/*| Class ---------------------------------------------------------------

   Name            : CsaGenericMain

   Description     : CsaGenericMain is a friend class of a couple of
                     basic libraries all designed using the singleton
                     pattern with a private constructor/destructor and
                     destroy() method. These singletons will be instantiated
                     in the open method of CsaGenericMain and destroyed
                     (in reverse order) in the close() method.

-----------------------------------------------------------------------------*/
class CsaGenericMain : public CsaGenericMainBas {

public:

  // Constructor
  CsaGenericMain(int argc, char *argv[])
```



```
    : CsaGenericMainBas(argc,argv) {}

  // The openHook() method is called by the svc() method and instantiates
  // all singletons the generic main provides for the component DLLs.
  virtual int openHook (void);

  // The closeHook() method is called by the svc() method to close up
  // all singletons the generic main provided for the components
  // (reverse order).
  virtual int closeHook (void);

  // Instances of of singletons
  CsaMscCommu    *commu_;
  CsaSesam       *sesam_;
  CsaOsSystem    *os_;
  CsaPmgtCmdIf   *rti_;

};

/*| END Class */


#endif  //ifndef CSAGENERICMAIN_H


/*| Change header -------------------------------------------------------

   Date         : 26. Jun 1996
   Version      : -
   Charm        : -
   Author       : D. Quehl (Qu); Siemens AG B Med CSA

   Description  : Initial release

-----------------------------------------------------------------------------*/
```

```
/*| Compilation unit ----------------------------------------------------

   Component       : CSA -

   Name            : CsaGenericMain.cpp

   Author          : (D. Quehl, BNEP11, +49 9131 84 2430); Siemens AG B Med CSA

   Language        : C/C++

   Creation Date   : 19-JUN-1996

   Test State      :

   Description     : Add on to Generic Main functionality.
                     Implementation of library startup/rundown functionality.

   Requirement Key : os_cs_process_mgt, os_os_test_testability

             Copyright (C) Siemens AG 1995 All Rights Reserved

-------------------------------------- ----------------------------------------*/
/*| END */

#pragma comment ( exestr, "%(#) CsaGenericMain.cpp 1.4 (10 Jul 1996)  test: Implementation Generic Main" )

#include <osc/CsaGenericMain.h>

/*| Method ------------------------------------------------------------------

   Name            : CsaGenericMain::openHook

   Description     : The openHook() method is called by the svc() method and
                     instantiates all singletons the generic main provides
                     for the component DLLs.

   Return          : 0                         success
                     -1                        error

-----------------------------------------------------------------------------*/
int CsaGenericMain::openHook (
                            void            // takes no arguments
                            )

/*| END Method */

{
  CSA_TRACE_IN ((CSA_OSC, "int CsaGenericMain::openHook"));

  this->sesam_ = CsaSesam::sesam();
  if (this->sesam_ == 0) {
    CSA_STATUS_GMAIN(NOMEM_NEW);
  }
  this->commu_ = CsaMscCommu::commu();
  if (this->commu_ == 0) {
```



```
    CSA_STATUS_GMAIN(NOMEM_NEW);
  }
  this->os_ = CsaOsSystem::csaOsSystem();
  if (this->os_ == 0) {
    CSA_STATUS_GMAIN(NOMEM_NEW);
  }
  this->rti_ = CsaPmgtCmdIf::csaPmgtCmdIf();
  if (this->rti_ == 0) {
    CSA_STATUS_GMAIN(NOMEM_NEW);
  }
  return 0;
}

/*| Method ------------------------------------------------------------------

   Name            : CsaGenericMain::closeHook

   Description     : The closeHook() method is called by the svc() method and
                     destroys all singletons the generic main provided for the
                     components (reverse order).

   Return          : 0                         success
                     -1                        error

-----------------------------------------------------------------------------*/
int CsaGenericMain::closeHook (
                            void            // takes no arguments
                            )

/*| END Method */

{
  // close down singletons

//  cout << "CsaGenericMain::closeHook>>> closing down libraries" << endl;

//  cout << "CsaPmgtCmdIf" << endl;
  this->rti_->destroy();

//  cout << "CsaOsSystem" << endl;
  this->os_->destroy();

//  cout << "CsaMsc" << endl;
  this->commu_->destroy();

//  cout << "CsaSesam" << endl;
  this->sesam_->destroy();


//  cout << "CsaGenericMain::closeHook>>> closing down libraries finished" << endl;

  return 0;
}

/*| Change header -----------------------------------------------------------
```

```
   Date        : 26. Jun 1996
   Version     : -
   Charm       : -
   Author      : D. Quehl (Qu); Siemens AG B Med CSA

   Description : Initial release

|-----------------------------------------------------------------------*/


/*| Change header -----------------------------------------------------

   Date        : 07. Jul 1996
   Version     : 1.1
   Charm       : 2041
   Author      : D. Quehl (Qu); Siemens AG B Med CSA

   Description : Changes in respect to code review component
                 "osc - generic main - 03.jul.96 "

|-----------------------------------------------------------------------*/
```